P@RTAL
USPTO

Search:　◉ The ACM Digital Library　○ The Guide

software project management microsoft project check points

THE ACM DIGITAL LIBRARY

Feedback　Report a problem　Satisfaction survey

Terms used
software project management microsoft project check points

Found **100,359** of **192,876**

Sort results by: relevance

Display results: expanded form

◆ Save results to a Binder
? Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200　　　　Result page: **1**　2　3　4　5　6　7　8　9　10　　next
Best 200 shown　　　　　　　　　　　　　　　　　　　　　Relevance scale ☐ ▭ ▩ ▩ ▩

1　Object database support for a software project management environment　　▩
Lung-Chun Liu, Ellis Horowitz
November 1988　**ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the third ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments SDE 3**, Volume 13 , 24 Issue 5 , 2
Publisher: ACM Press
Full text available: 📄 pdf(1.39 MB)　　Additional Information: full citation, abstract, references, index terms

> The recent development of object-oriented database models, which combine the power of object programming and the efficient management of data, provides a feasible solution for the construction of a computer-aided software engineering environment or CASE. However, an object oriented database provides only a kernel set of capabilities. This paper identifies the data management requirements related to software project management and shows how they are represented in the model called Design-Net ...

2　Papers: Tracking software projects with the integrated version control in SMIT　　▩
Zorica Mihajlovic, Dusan Velasevic
March 2001　**ACM SIGSOFT Software Engineering Notes**, Volume 26 Issue 2
Publisher: ACM Press
Full text available: 📄 pdf(781.64 KB)　Additional Information: full citation, abstract, references

> The purpose of the software project tracking and oversight process is to provide visibility into actual progress so that management can take corrective actions when the project's performance deviates significantly from the plans. The sequence of the project's performances forms the project's change history. The new SMIT tool for planning and tracking software projects with the additional capability to process the project's change history is presented in the paper. This additional capability is b ...

> **Keywords**: project tracking, software projects, tools, version control

3　Frontmatter (TOC, Letters, Philosophy of computer science, Interviewers needed,　▩
Taking software requirements creation from folklore to analysis, SW components and product lines: from business to systems and technology, Software engineering survey)
September 2005　**ACM SIGSOFT Software Engineering Notes**, Volume 30 Issue 5
Publisher: ACM Press
Full text available: 📄 pdf(1.98 MB)　　Additional Information: full citation, index terms

4   Impact of software engineering research on the practice of software configuration
management
Jacky Estublier, David Leblang, André van der Hoek, Reidar Conradi, Geoffrey Clemm, Walter
Tichy, Darcy Wiborg-Weber
October 2005 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 14 Issue 4
Publisher: ACM Press
Full text available: pdf(350.59 KB)   Additional Information: full citation, abstract, references, index terms

> Software Configuration Management (SCM) is an important discipline in professional
> software development and maintenance. The importance of SCM has increased as
> programs have become larger, more long lasting, and more mission and life critical. This
> article discusses the evolution of SCM technology from the early days of software
> development to the present, with a particular emphasis on the impact that university and
> industrial research has had along the way. Based on an analysis of the publicati ...
>
> **Keywords**: Versioning, data model, process support, research impact, software
> configuration management, software engineering, workspace management

5   Agent based modeling: agent-based simulation in AI planning: Agent-based
simulation for software project planning
David Joslin, William Poole
December 2005 **Proceedings of the 37th conference on Winter simulation WSC '05**
Publisher: Winter Simulation Conference
Full text available: pdf(101.00 KB)   Additional Information: full citation, abstract, references

> Estimates of task duration and resource requirements in software engineering are
> notoriously inaccurate, and as a result effective project management often must be very
> dynamic. In response to new information or revised estimates, it may be necessary to
> reassign resources, cancel optional tasks, etc. Project management tools that make
> projections while treating decisions about tasks and resource assignments as static will
> not yield realistic results. In this paper we describe some preliminary at ...

6   Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced**
Studies on Collaborative research
Publisher: IBM Press
Full text available: pdf(4.21 MB)   Additional Information: full citation, abstract, references, index terms

> Understanding distributed applications is a tedious and difficult task. Visualizations based
> on process-time diagrams are often used to obtain a better understanding of the
> execution of the application. The visualization tool we use is Poet, an event tracer
> developed at the University of Waterloo. However, these diagrams are often very complex
> and do not provide the user with the desired overview of the application. In our
> experience, such tools display repeated occurrences of non-trivial commun ...

7   Special issue on persistent object systems: Orthogonally persistent object systems
Malcolm Atkinson, Ronald Morrison
July 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**,
Volume 4 Issue 3
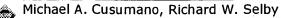Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(5.02 MB)   Additional Information: full citation, abstract, references, citings

> Persistent Application Systems (PASs) are of increasing social and economic importance.
> They have the potential to be long-lived, concurrently accessed, and consist of large
> bodies of data and programs. Typical examples of PASs are CAD/CAM systems, office
> automation, CASE tools, software engineering environments, and patient-care support
> systems in hospitals. Orthogonally persistent object systems are intended to provide

improved support for the design, construction, maintenance, and operation o ...

Keywords: database programming languages, orthogonal persistence, persistent application systems, persistent programming languages

8 How Microsoft builds software
Michael A. Cusumano, Richard W. Selby
June 1997 **Communications of the ACM**, Volume 40 Issue 6
Publisher: ACM Press

Full text available: pdf(712.71 KB)   Additional Information: full citation, references, citings, index terms, review

9 Courses: State of the art in interactive ray tracing
Peter Shirley
July 2006 **Material presented at the ACM SIGGRAPH 2006 conference SIGGRAPH '06**
Publisher: ACM Press

Full text available: pdf(14.08 MB)   Additional Information: full citation, abstract

Recent improvements in computer hardware have allowed ray tracing to be used in some interactive applications. The trends in architecture and expansions of geometric model should increase the use of interactive ray tracing. This course presents recent and often not-yet published work on interactive ray tracing.

10 Daily build and feature development in large distributed projects
Even-André Karlsson, Lars-Göran Andersson, Per Leion
June 2000 **Proceedings of the 22nd international conference on Software engineering**
Publisher: ACM Press

Full text available: pdf(93.48 KB)   Additional Information: full citation, abstract, references, citings, index terms

Daily build is a software development paradigm that originated in the PC industry to get control of the development process, while still allowing the focus on end user requirements and code. The PC industry used daily build to avoid chaos in increasingly larger applications in an environment without a strong development process. Ericsson Radio Systems has chosen to implement daily build to increase the focus on end user requirements and code, but from a different starting point with a tradi ...

Keywords: daily build, extreme programming, feature teams, incremental development, integrated product teams

11 Frontmatter (TOC, Letters, Election results, Software Reliability Resources!, Computing Curricula 2004 and the Software Engineering Volume SE2004, Software Reuse Research, ICSE 2005 Forward)
July 2005 **ACM SIGSOFT Software Engineering Notes**, Volume 30 Issue 4
Publisher: ACM Press
Full text available: pdf(6.19 MB)   Additional Information: full citation, index terms

12 GPGPU: general purpose computation on graphics hardware
David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press

Full text available: 📄 pdf(63.03 MB)   Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

13 Invited papers: Impact of the research community on the field of software configuration management: summary of an impact project report

Jacky Estublier, David Leblang, Geoff Clemm, Reidar Conradi, Walter Tichy, André van der Hoek, Darcy Wiborg-Weber

September 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 5

Publisher: ACM Press

Full text available: 📄 pdf(1.22 MB)   Additional Information: full citation, abstract, references, citings

Software Configuration Management (SCM) is an important discipline in professional software development and maintenance. The importance of SCM has increased as programs have become larger and more complex and mission/life-critical. This paper discusses the evolution of SCM technology from the early days of software development to present and the impact university and industrial research has had along the way. It also includes a survey of the industrial state-of-the-practice and research directio ...

Keywords: industrial impact, software configuration management, software engineering, software quality

14 Courses: An introduction to sketch-based interfaces

Joseph LaViola, Randall Davis, Takeo Igarashi

July 2006 **Material presented at the ACM SIGGRAPH 2006 conference SIGGRAPH '06**

Publisher: ACM Press

Full text available: 📄 pdf(31.58 MB)   Additional Information: full citation, abstract

Sketch-based interfaces are a natural, pencil-and-paper-like approach to interacting with a variety of applications, including conceptual modeling, animation, and note-taking systems. This course offers an in-depth discussion of sketch-based interface design, ranging from simple gestural commands to complex sketch-understanding systems. Attendees will learn how these interfaces are designed and how to develop their own.

15 Level II technical support in a distributed computing environment

Tim Leehane

September 1996 **Proceedings of the 24th annual ACM SIGUCCS conference on User services**

Publisher: ACM Press

Full text available: 📄 pdf(5.73 MB)   Additional Information: full citation, references, index terms

16 An analysis of XML database solutions for the management of MPEG-7 media descriptions

Utz Westermann, Wolfgang Klas

December 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 4

Publisher: ACM Press

Full text available: 📄 pdf(448.76 KB)   Additional Information: full citation, abstract, references, index terms, review

MPEG-7 constitutes a promising standard for the description of multimedia content. It can be expected that a lot of applications based on MPEG-7 media descriptions will be set up in the near future. Therefore, means for the adequate management of large amounts of MPEG-7-compliant media descriptions are certainly desirable. Essentially, MPEG-7 media

descriptions are XML documents following media description schemes defined with a variant of XML Schema. Thus, it is reasonable to investigate curren ...

Keywords: MPEG-7, XML database systems, multimedia databases

17 Coordination: Software configuration management over a global software development environment: lessons learned from a case study
Leonardo Pilatti, Jorge Luis Nicolas Audy, Rafael Prikladnicki
May 2006 **Proceedings of the 2006 international workshop on Global software development for the practitioner GSD '06**
Publisher: ACM Press
Full text available: pdf(234.54 KB)   Additional Information: full citation, abstract, references, index terms

Software configuration management is an important support activity in the software development process. In global environments, the software configuration becomes critical due to the characteristics of the distributed development (physical distance, cultural differences, trust, communication and other factors). The objective of this paper is to analyze the software configuration management in a global software development environment, identifying the main challenges. The results are based on a c ...

Keywords: global software development, software configuration management, software process improvement

18 Real-time shading
Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(7.39 MB)   Additional Information: full citation, abstract

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders of thousands to tens of thousands of instructions. This course has been redesigned to address today's real-time shading capabili ...

19 Reports: A historical perspective on runtime assertion checking in software development
Lori A. Clarke, David S. Rosenblum
May 2006 **ACM SIGSOFT Software Engineering Notes**, Volume 31 Issue 3
Publisher: ACM Press
Full text available: pdf(473.68 KB)   Additional Information: full citation, abstract, references, index terms

This report presents initial results in the area of software testing and analysis produced as part of the Software Engineering Impact Project. The report describes the historical development of runtime assertion checking, including a description of the origins of and significant features associated with assertion checking mechanisms, and initial findings about current industrial use. A future report will provide a more comprehensive assessment of development practice, for which we invite readers ...

20 Assessing process-centered software engineering environments
Vincenzo Ambriola, Reidar Conradi, Alfonso Fuggetta
July 1997 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 6 Issue 3
Publisher: ACM Press
Full text available: pdf(342.52 KB)   Additional Information: full citation, abstract, references, citings, index terms, review

Process-centered software engineering environments (PSEEs) are the most recent generation of environments supporting software development activities. They exploit an representation of the process (called the process model that specifies how to carry out software development activities, the roles and tasks of software developers, and how to use and control software development tools. A process model is therefore a vehicle to better understand and communicate the process. If ...

Keywords: CASE, enabling technology, process modeling languages, process-centered software engineering environments, software process

Results 1 - 20 of 200   Result page: **1**   2   3   4   5   6   7   8   9   10   next

PORTAL

USPTO

Search: ● The ACM Digital Library   O The Guide

sequential software development software project managemer

THE ACM DIGITAL LIBRARY

❦ Feedback Report a problem Satisfaction survey

Terms used
sequential software development software project management

Found **100,052** of **192,876**

Sort results by   [relevance ▼]
Display results   [expanded form ▼]

● Save results to a Binder
? Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200
Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10   next

Relevance scale ☐ ▭ ▥ ▦ ▩

1 Issues encountered in building a flexible software development environment: lessons from the Arcadia project
R. Kadia
November 1992 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fifth ACM SIGSOFT symposium on Software development environments SDE 5**, Volume 17 Issue 5
Publisher: ACM Press

Full text available: 📄 pdf(1.51 MB)   Additional Information: full citation, abstract, references, citings, index terms

This paper presents some of the more significant technical lessons that the Arcadia project has learned about developing effective software development environments. The principal components of the Arcadia-1 architecture are capabilities for process definition and execution, object management, user interface development and management, measurement and evaluation, language processing, and analysis and testing. In simultaneously and cooperatively developing solutions in these areas we learned ...

2 Concurrent software development
Joseph Blackburn, Gary Scudder, Luk N. Van Wassenhove
November 2000 **Communications of the ACM**
Publisher: ACM Press

Full text available: 📄 pdf(577.12 KB)   Additional Information: full citation, references, index terms

3 Keynote talks: A view of 20th and 21st century software engineering
Barry Boehm
May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06**
Publisher: ACM Press

Full text available: 📄 pdf(329.46 KB)   Additional Information: full citation, abstract, references, index terms

George Santayana's statement, "Those who cannot remember the past are condemned to repeat it," is only half true. The past also includes successful histories. If you haven't been made aware of them, you're often condemned not to repeat their successes.In a rapidly expanding field such as software engineering, this happens a lot. Extensive studies of many software projects such as the Standish Reports offer convincing evidence that many projects fail to repeat past successes.This paper tries to i ...

Keywords: software engineering, software futures, software history

4 Economics, management and mathematics: Strategizing software development: strategic management of internet service development

Masao Kakihara

May 2006 **Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research WISER '06**

Publisher: ACM Press

Full text available: pdf(392.07 KB)   Additional Information: full citation, abstract, references, index terms

> This paper explores a theoretical linkage between software engineering and strategic management. Software engineering is now faced with two dynamic innovation streams: technological innovation and market innovation. Harshly shook by rapid technological development and highly volatile market environments, today's software development is under the constant necessity for swift and reliable development practices and market launch in appropriate timing. In short, software development has to be more a ...

> **Keywords:** internet service, software development, strategic management

5 Abstraction-based software development

Valdis Berzins, Michael Gray, David Naumann

May 1986 **Communications of the ACM,** Volume 29 Issue 5

Publisher: ACM Press

Full text available: pdf(1.65 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

> A five-year experience with abstraction-based software-development techniques in the university environment indicates that the investment required to support the paradigm in practice is returned in terms of greater ability to control complexity in large projects—provided there exists a set of software tools sufficient to support the approach.

6 An object-oriented model of software configuration management

Hal Render, Roy Campbell

May 1991 **Proceedings of the 3rd international workshop on Software configuration management**

Publisher: ACM Press

Full text available: pdf(1.21 MB)   Additional Information: full citation, references, citings, index terms

7 Impact of software engineering research on the practice of software configuration management

Jacky Estublier, David Leblang, André van der Hoek, Reidar Conradi, Geoffrey Clemm, Walter Tichy, Darcy Wiborg-Weber

October 2005 **ACM Transactions on Software Engineering and Methodology (TOSEM),** Volume 14 Issue 4

Publisher: ACM Press

Full text available: pdf(350.59 KB)   Additional Information: full citation, abstract, references, index terms

> Software Configuration Management (SCM) is an important discipline in professional software development and maintenance. The importance of SCM has increased as programs have become larger, more long lasting, and more mission and life critical. This article discusses the evolution of SCM technology from the early days of software development to the present, with a particular emphasis on the impact that university and industrial research has had along the way. Based on an analysis of the publicati ...

> **Keywords:** Versioning, data model, process support, research impact, software configuration management, software engineering, workspace management

8

Models of software development environments

D. E. Perry, G. E. Kaiser

April 1988 **Proceedings of the 10th international conference on Software engineering**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.11 MB)   Additional Information: full citation, abstract, references, citings, index terms

We present a general model of software development environments that consists of three components: policies, mechanisms and structures. The advantage of this formalization is that it distinguishes precisely those aspects of an environment that are useful in comparing and contrasting software development environments. We introduce four classes of models by means of a sociological metaphor that emphasizes scale: the individual, the family, the city and the state models. The utility of this ta ...

9 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix

March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Publisher: ACM Press

Full text available: pdf(7.97 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

10 On the use of formal methods in software development

D. Bjorner

March 1987 **Proceedings of the 9th international conference on Software Engineering**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.15 MB)   Additional Information: full citation, abstract, references, citings, index terms

We propose a total framework for the software development stages of specification (definition), design and coding. This framework is based on three cornerstones: (a) the concept of software development graphs which specify all the stages and steps of development; (b) the use of formal methods, in our case VDM, the Vienna Software Development Method, in all stages and steps of development; and (c) the clearly separate rôles of theoretical computer scientists, programmers, software engi ...

11 Workshop on Open Source Software Engineering (WOSSE): Observations on patterns of development in open source software projects

Katherine J. Stewart, David P. Darcy, Sherae L. Daniel

May 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fifth workshop on Open source software engineering 5-WOSSE**, Volume 30 Issue 4

Publisher: ACM Press

Full text available: pdf(187.59 KB)   Additional Information: full citation, abstract, references, index terms

This paper discusses a project aimed at understanding how open source software evolves by examining patterns of development and changes in releases over time. The methodological approach of the research and initial observations are described. These include descriptions of release cycles and categorization of projects based on the overall changes in size and complexity exhibited across releases. Implications of these observations are discussed in light of prior and future work on understanding OS ...

Keywords: open source software, software evolution

12

Sequential patterns in information systems development: an application of a social process model

Daniel Robey, Michael Newman

January 1996 **ACM Transactions on Information Systems (TOIS)**, Volume 14 Issue 1

Publisher: ACM Press

Full text available: pdf(2.38 MB)

Additional Information: full citation, abstract, references, citings, index terms, review

We trace the process of developing and implementing a materials management system in one company over a 15-year period. Using a process research model developed by Newman and Robey, we identify 44 events in the process and define them as either encounters or episodes. Encounters are concentrated events, such as meetings and announcements, that separate episodes, which are events of longer duration. By examining the sequence of events over the 15 years of the case, we identify a pattern of r ...

Keywords: social processes, system implementation

13 Parallel changes in large-scale software development: an observational case study

Dewayne E. Perry, Harvey P. Siy, Lawrence G. Votta

July 2001 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 10 Issue 3

Publisher: ACM Press

Full text available: pdf(361.44 KB)

Additional Information: full citation, abstract, references, citings, index terms, review

An essential characteristic of large-scale software development is parallel development by teams of developers. How this parallel development is structured and supported has a profound effect on both the quality and timeliness of the product. We conduct an observational case study in which we collect and analyze the change and configuration management history of a legacy system to delineate the boundaries of, and to understand the nature of, the problems encountered in parallel development. ...

Keywords: change management, merging interfering and noninterfering versions, parallel versions, parallel/concurrent changes, software integration

14 A reusable, academic-strength, metrics-based software engineering process for capstone courses and projects

Richard Conn

March 2004 **ACM SIGCSE Bulletin , Proceedings of the 35th SIGCSE technical symposium on Computer science education SIGCSE '04**, Volume 36 Issue 1

Publisher: ACM Press

Full text available: pdf(223.31 KB)

Additional Information: full citation, abstract, references, citings, index terms

This paper describes a mature Software Engineering Process that may be applied to capstone courses, student projects, and research projects in a university environment. This process, based in part on the Team Software Process of the Software Engineering Institute, features mature software engineering best practices, including extensive use of metrics to gain insight into process effectiveness and product quality. It is designed to be executed in a single 16-week semester, and it can easily be mo ...

Keywords: capability maturity model, communicating sequential process, data primitives, defect, personal software process, process improvement, software engineering institute, structured query language, team software process, unified modeling language, visual basic

15 Version models for software configuration management

Reidar Conradi, Bernhard Westfechtel
June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2
Publisher: ACM Press

Full text available: pdf(483.54 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

> After more than 20 years of research and practice in software configuration management (SCM), constructing consistent configurations of versioned software products still remains a challenge. This article focuses on the version models underlying both commercial systems and research prototypes. It provides an overview and classification of different versioning paradigms and defines and relates fundamental concepts such as revisions, variants, configurations, and changes. In particular, we foc ...
>
> **Keywords**: changes, configuration rules, configurations, revisions, variants, versions

16 Communication metrics for software development

Bernd Bruegge, Allen H. Dutoit
May 1997 **Proceedings of the 19th international conference on Software engineering**
Publisher: ACM Press

Full text available: pdf(1.74 MB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>index terms</u>

> **Keywords**: communicication, empirical software engineering, software development, structural equations, team-based projects

17 Software engineering-as it is

Barry W. Boehm
September 1979 **Proceedings of the 4th international conference on Software engineering**
Publisher: IEEE Press

Full text available: pdf(1.01 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

> This paper presents a view of software engineering as it is in 1979. It discusses current software engineering practice with respect to lessons learned in the past few years, and concludes that the lessons are currently not heeded roughly half of the time. The paper discusses some of the factors which may account for this lag, including rapid technological change, education shortfalls, technology transfer inhibitions, resistance to disciplined methods, inappropriate role models, and a restr ...

18 How Microsoft builds software

Michael A. Cusumano, Richard W. Selby
June 1997 **Communications of the ACM**, Volume 40 Issue 6
Publisher: ACM Press

Full text available: pdf(712.71 KB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

19 Dynamic software engineering: An evolutionary approach to automated software development and management

Joseph S. Greene
October 1976 **Proceedings of the 2nd international conference on Software engineering**
Publisher: IEEE Computer Society Press

Full text available: pdf(721.47 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

> Cybernomorphic leverage techniques have been extracted from the Air Force Weapons

Laboratory HULL system and generalized to provide a new management system called SAIL for producing, updating and maintaining software and data files. SAIL is being used on CDC6600, CDC7600, IBM 360/85 and HIS6080 computers at twelve installations. The system provides a totally new level of centralized software development and maintenance control, while simultaneously decentralizing user applications, providin ...

Keywords: Cybernomorphic, FIVTRAN, Software engineering, Strategic planning

20 Information and process quality: The IBM-McGill project on software process
Nazim H. Madhavji, Kamel Toubache, Ed Lynch
October 1991 **Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research**
Publisher: IBM Press
Full text available: pdf(1.02 MB)   Additional Information: full citation, abstract, references

Historically, the process used to develop software has played an important role in the field of software engineering. A number of software lifecycle models have been developed in the last three decades. These models, while helpful in giving general guidance to software developers, do not expose the myriad details that are critical in any large, evolving software development project. Recent developments, however, have unfolded many hidden aspects of the software process giving rise to a new disci ...

Results 1 - 20 of 200   Result page: **1**   2   3   4   5   6   7   8   9   10   next